



University of Dhaka

Department of Computer Science and Engineering

CSE 3113: Software Engineering

Software Requirements Specification
VocalPrep, An Ai powered Mock Interviewer

Version 1.0

Lab Group: A-?

Submitted by:

Sakib Hossain (05)

Faria Yasmin (21)

Yuki Bhuiyan (64)

Submitted To:

Dr. Sarkar Tanveer Ahmed Rume, Associate Professor, Dept. of CSE,
University of Dhaka

Dr. Saifuddin Md. Tareeq, Professor, Dept. of CSE, University of Dhaka

Submission Date: 2nd July, 2025

Revision History

Date	Version	Description	Author
2/7/2025	1.0	Initial draft of SRS document with complete functional and non-functional requirements	Group A

Table of Contents

Table of Contents	3
Software Requirements Specification: AI Interview Prep Platform	4
1. Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions, Acronyms and Abbreviations	5
1.4 Overview	6
2. Overall Description	6
2.1 Use-Case Model Survey	6
2.2 Assumptions and Dependencies	7
3. Specific Requirements	8
3.1 Use-Case Reports	8
3.2 Supplementary Requirements	11
4. Reference	

Software Requirements Specification: AI Interview Prep Platform

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the requirements for **VocalPrep**, an **AI Interview Preparation Platform**. This SRS fully describes the external behavior of the platform, which enables users to practice for job interviews. It specifies the functional requirements, such as user authentication, custom interview creation, and AI-driven feedback, as well as nonfunctional requirements, including performance, security, and usability, which are necessary to provide a complete and comprehensive description of the software.

1.2 Scope

This SRS applies to the development of the AI Interview Prep Platform. The platform's core features include:

- User authentication managed via the Clerk service.
- The ability for users to create customized interview sessions by selecting a professional field, the number of questions, and AI model preferences.
- AI-powered analysis of user responses using the Google Gemini AI model.
- Delivery of feedback through a text-based report and an optional Text-to-Speech (TTS) feature.

- Secure storage of user data, including interview history and progress, using the Firebase Realtime Database.
- A responsive and modern User Interface (UI) built with Shadcn components.

This document is associated with the primary Use-Case model for the platform, which covers all user-facing interactions.

1.3 Definitions, Acronyms and Abbreviations

- **AI:** Artificial Intelligence
- **API:** Application Programming Interface
- **Clerk:** A third-party service for user authentication and management.
- **Firebase:** A platform by Google for creating mobile and web applications, used here for its Realtime Database.
- **Gemini:** A family of multimodal AI models developed by Google.
- **SRS:** Software Requirements Specification
- **TTS:** Text-to-Speech
- **UI:** User Interface
- **Shadcn:** A collection of re-usable UI components for React/Next.js applications.

1.4 Overview

This document is organized into three main sections. Section 1 provides a general introduction to the project and this document. Section 2 offers an overall description of the product, its users, and its operational environment, including a survey of the use-case model. Section 3 details the specific functional and non-functional requirements. Functional requirements are defined through use-case reports, and non-functional requirements are listed as supplementary requirements.

2. Overall Description

2.1 Use-Case Model Survey

The system provides a set of functionalities for a single primary actor: the **User** (e.g., a job candidate).

Actor:

- **User:** An individual who wants to practice for job interviews.

Use-Case Diagram Overview:

The User actor can perform the following primary use cases: Register and Log In, Create Custom Interview, Start and Complete Interview Session, Receive AI-Powered Feedback, and View Interview History.

Use Cases:

- **UC-01: Manage Account (Register & Login):** The user registers for a new account or logs into an existing one to access the platform.
- **UC-02: Create Custom Interview:** The user defines the parameters for a new mock interview, including the job field, number of questions, and other preferences.
- **UC-03: Start and Complete Interview Session:** The user participates in the mock interview, answering questions presented by the system.
- **UC-04: Receive AI-Powered Feedback:** After an interview, the user receives a detailed analysis of their performance.
- **UC-05: View Interview History:** The user can access a list of their past interviews and review the associated feedback.

2.2 Assumptions and Dependencies

- **Assumptions:**
 - Users have a stable internet connection and a modern web browser with JavaScript enabled.

- Users will have a functional microphone for providing spoken answers during the interview session.
 - The content for job fields and a base pool of interview questions will be pre-populated or managed via a simple administrative interface (not in scope for this SRS).
-
- **Dependencies:**
 - The system is critically dependent on the availability and performance of third-party services:
 - **Clerk:** For all user authentication and session management. Any downtime will prevent user access.
 - **Google Gemini API:** For all AI-driven analysis of interview responses. API latency or downtime will impact the feedback generation feature.
 - **Firebase Realtime Database:** For all user data storage. Downtime will affect data persistence and retrieval.
 - The front-end is dependent on the **TypeScript** language and UI components from the **Shadcn** library.
 - The browser's native **Web Speech API** or a similar library will be used for speech-to-text and text-to-speech functionalities.

3. Specific Requirements

3.1 Use-Case Reports

UC-01: Manage Account (Register & Login)

- **Description:** The user creates a new account or logs into the system using the Clerk authentication interface.
- **Flow of Events:**
 1. The user navigates to the application's landing page.
 2. The user clicks "Sign Up" or "Log In".
 3. The system presents the Clerk-provided authentication modal.
 4. For registration, the user provides credentials (e.g., email, password, or social provider). For login, the user enters existing credentials.
 5. Clerk validates the credentials and authenticates the user.
 6. Upon successful authentication, the user is redirected to their personal dashboard.

UC-02: Create Custom Interview

- **Description:** The user configures a new mock interview session.
- **Flow of Events:**
 1. From the dashboard, the user selects an option to "Create a New Interview".
 2. The system displays a configuration form.
 3. The user selects a job field from a predefined list (e.g., "Software Engineering," "Marketing").
 4. The user specifies the number of questions for the interview.
 5. The user can set AI preferences (e.g., difficulty level, feedback style).

6. The user saves the configuration, which is stored in the Firebase database and linked to their user profile.

UC-03: Start and Complete Interview Session

- **Description:** The user engages in the configured mock interview.
- **Flow of Events:**
 1. The user selects a previously created interview from their dashboard to start the session.
 2. The system presents the first interview question, both as text on the screen and optionally via TTS.
 3. The user provides their answer verbally. The system records the audio and transcribes it to text.
 4. The user indicates they have finished their answer and proceeds to the next question.
 5. Steps 3-4 are repeated until all questions are answered.
 6. Upon completion, the system informs the user that their answers are being analyzed.

UC-04: Receive AI-Powered Feedback

- **Description:** The user reviews the AI-generated analysis of their interview performance.
- **Flow of Events:**
 1. After the interview is completed and analyzed, the user is navigated to a feedback page.
 2. The system sends the transcribed answers to the Google Gemini API for evaluation.
 3. The Gemini API returns a structured analysis (e.g., strengths, areas for improvement, ratings on specific criteria).

4. The system displays this analysis in a clear, readable format.
5. The user has an option to listen to the feedback via a TTS feature.
6. This feedback report is saved to Firebase as part of the interview's history.

UC-05: View Interview History

- **Description:** The user accesses and reviews their past interview sessions and feedback.
- **Flow of Events:**
 1. The user navigates to their dashboard or an "Interview History" section.
 2. The system retrieves and displays a list of all completed interviews from Firebase, showing key details like date and job field.
 3. The user can click on any past interview to view the full feedback report associated with it.

3.2 Supplementary Requirements

SR-1: Usability and User Interface

- **SR-1.1:** The UI shall be built using Shadcn components to ensure a modern, clean, and consistent look and feel.
- **SR-1.2:** The application shall be fully responsive, providing an optimal viewing experience across a wide range of devices, including desktops, tablets, and mobile phones.
- **SR-1.3:** Navigation shall be intuitive, allowing users to access all major features (Create Interview, History, Logout) from a primary navigation menu or dashboard.

SR-2: Performance

- **SR-2.1:** Page load times for the main dashboard and interview setup pages shall be under 3 seconds on a standard broadband connection.
- **SR-2.2:** The AI feedback generation process, from submission of answers to display of results, should complete within 45 seconds to maintain user engagement.
- **SR-2.3:** The TTS playback shall start within 2 seconds of the user's request.

SR-3: Security

- **SR-3.1:** All user authentication, including password storage and session management, shall be delegated to and handled by Clerk. The application will not store raw passwords.
- **SR-3.2:** All communication between the client and server, as well as with third-party APIs (Clerk, Gemini, Firebase), must be encrypted using HTTPS.
- **SR-3.3:** Firebase security rules shall be configured to ensure that a user can only access and modify their own data (interview configurations, history, and feedback).

SR-4: Reliability

- **SR-4.1:** The system's uptime is dependent on the uptime of its core external services (Clerk, Google Cloud, Firebase). The application should handle API failures gracefully, informing the user of any temporary service unavailability.

SR-5: Technology Stack

- **SR-5.1:** The application logic and front-end shall be written in **TypeScript** to ensure type safety and maintainability.
- **SR-5.2:** The backend data persistence layer shall be implemented using **Firebase Realtime Database**.
- **SR-5.3:** AI-powered analysis must be performed by the **Google Gemini** model via its official API.

References

https://sceweb.uhcl.edu/helm/RUP_Folder/RationalUnifiedProcess/webtmpl/template_s